

Comparison of Different Software Architectures for Data Integration

Yasin Osman Mohammed¹
e-mail: yasmath1@gmail.com

Roliana Ibrahim²
e-mail: roliana@utm.my

Author(s) Contact Details:

¹ University of Khartoum, Khartoum, Sudan

² Universiti Teknologi Malaysia, Johor Bahru, Malaysia

Abstract— As the result of continuous development and the emergence of new technologies in the field of information systems, the issues of data integration has introduced a lot of concern for the past decades. Using heterogeneous systems and applications to support variety of processes has become the current trend in the adoption of information technology by organizations. From a business point of view, most of the organizational business processes are interrelated in some way. As such, the systems which have been implemented by organizations to perform various and multiple operations, should be able to interact and exchange data seamlessly to maximise productivities. One of the most used approach in conducting data integration is through building a data center. The purpose of a data centre is to allow data exchange among different types of applications. The purpose of this paper is to conduct a systematic literature review, in order to compare different structures and technologies used by data centers, such as the structures and technologies for data integration access, distributed real-time union view, multi-agent's social, and service oriented architecture. The paper also reveals how the multilayer software architecture could increase the flexibility of a software and facilitate the integration post-processes for organizations.

Keywords - datacenter, enterprise application integration, heterogeneous, software architecture, data exchange

1. INTRODUCTION

Due to continuous increment of complex business requirements, the traditional structure of software systems and the way of developing these systems has changed and facing with several limitations. Modern kind of software architectures has the ability to facilitate integration processes and support interoperability between these systems. Various technologies have been used to produce the plug and play components such as Common Object Request Broker Architecture (CORBA) and Service-Oriented Architecture (SOA) for designing and developing softwares in the form of interoperable services. [1].

With those plug and play technologies and components, organizations keeps on adopting them to support their business processes. Nowadays even small organizations are able to run their operations with different small applications supporting the majority of their business processes. One way an organization can achieve a high efficiency of its business process is through improving information exchange among the applications. This approach is known as Enterprise Application Integration (EAI) [2].

Many approaches have adopted EAI, in order to achieve common user interface, integration by applications and middleware. However, the most used technique in EAI is the implementation of data center, which provides a unified aggregation point that all applications can communicate and exchange data with each other. The aim of this paper is to provide the results of the investigation and reviews among different technologies and approaches that have been adopted in the implementation of data centers. The paper also explain some advantages and limitation of each technology and their suitability for data integration through answering the following questions.

- 1) What are the challenges of integrating and exchanging data among enterprise applications?
- 2) What are the characteristics of software architecture that overcome those challenges and facilitate the software interoperability?

2. DATA INTEGRATION

Over a few years ago, the quality of information structure has been enhanced greatly. The database software has become much better as the information resources are aggregated to be used by the systems having been used for many decades. These enhancements have become precious data resources and software prosperity [4]. Now, the main inconveniences are the heterogeneity of systems leading to many "information isolated islands" which can't interact with each other. The sensibilities of the different analysis backgrounds and time of the primary distinct types of each system lead to isomerism, such as distinctive software and hardware environment, different development tools, unlike period designs and different development styles. This isolation caused the problem of lack system's expansibility, openness and the ability to interact with other systems. The distinct system's objectives leads to overlap and redundancy of the data, the application which allocated by different departments couldn't share their data, which increase the data entry, decreases the performance and creates a huge problems of essential applications [5]. Altogether, these problems critically bound the scope and performance of all-in-

one structure. Furthermore, these problems extremely obstruct the process of development which increase the needs of data integration [6]. Many approaches have been used for data integration such as integration by applications where the applications access various data sources and return integrated results to the user, Integration by middleware here the middleware provides reusable functionality that is generally used to solve dedicated aspects of the integration problem, e.g., As done by SQL-middleware “uniform data access” in this case, a logical integration of data is accomplished at the data access level [7]. This approach of integration has gotten a lot of concerns last years as the other technology models has been used. So before going into details of these technologies following is some of the common challenges faces the organizations in term of performing the data integration

- ***Data integration challenges***

In order to clarify the obstacles and the barriers that face the data integration in data centers, a study has been conducted by the Progress Software Corporation (NASDAQ: PRGS is a global software company that enables enterprises to be operationally responsive to changing conditions and customer interactions as they occur). This study reveals out continuous increasing in a research in the field of data integration. Regarding the past two years the study show that 55% of the respondents had performed more than integration efforts and the most of respondents do not think this number will decrease soon, but in opposite 48% predicted an increment of the number of integration efforts during next two years, whereas 41% said it would stay the same. Furthermore the study highlighted the most critical issues in data integration faces this organization as illustrated in figure1 Respondents pointed out multiple barriers to successful integration.

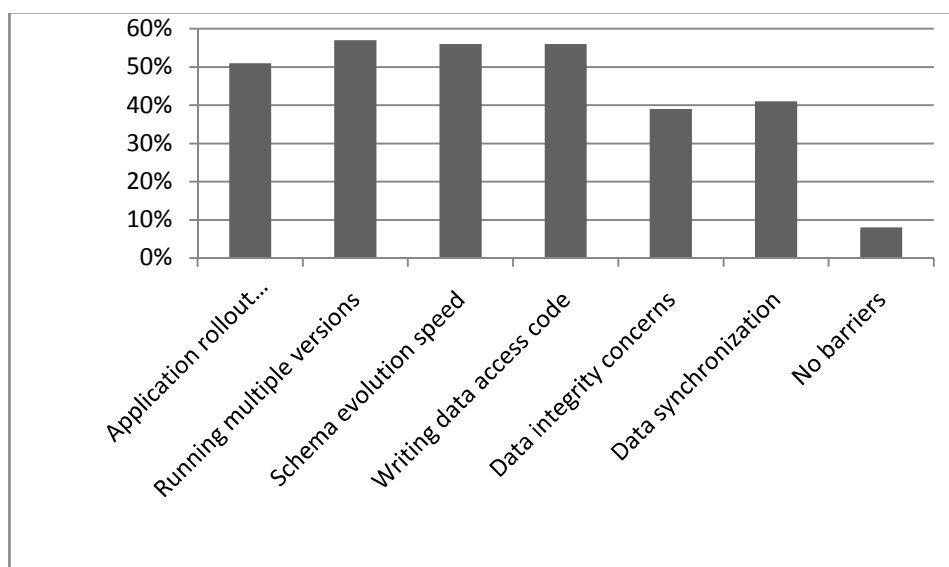


FIGURE 1: The barriers and challenges face the data integration [3]

The application rollout issue acquired the concern of 51%, and 57% reported the problem of running different versions of their applications simultaneously using the same database. 56% reported that the need for continuous changes to database schema slows down the application development processes, and 56% also reported the needs of rewriting or modifying the code each time of launching a new application or version.

Another study prepared by the Health Industry Insights (an advisory services and market research firm that emphasis on developing and employing strategies that leverage IT investments to maximize organizational performance in the field of healthcare industry) [8]. The purpose of this research was to highlight the critical system and data integration challenges faced by these provider organizations, the criteria for their selection of an integration platform, and the benefits they have realized from its implementation. By conducting an interview with the senior IT executives and system architects at four hospitals acknowledged to be industry Leaders in their use of HIT. The finding of the study is:

- Constrained interface development teams could not keep up with the sheer volume of interface activity to connect both internal and external systems.
- Aging, legacy integration engine technology did not provide a sufficiently rapid development environment.
- The product life cycle for an incumbent vendor resulted in too many costly and disruptive product upgrades.

- Interface volumes are increasing significantly—especially to connect external stakeholders such as community-based physician practices.
- Aging legacy interface engines cannot keep pace with this growing demand and the complexity of heterogeneous IT environments created through mergers and acquisitions.

Following is a discussion of a different data center model, design, technology and how it overcomes data integration challenges.

3. DATA CENTRE AS A TECHNIQUE FOR DATA INTEGRATION

Firstly it is important to indicate that the word “data center” used here metaphorically to describe the process of collecting the data from different information systems and expose it in a specific format and specific syntax “data integration”. The structure, design and the arrangement of the data in the data center responds to specific situations.

A. Basic plan of the data center

There are many factors used to determine the suitable plan of data center such as developing environment, system’s architecture, data management process, the required level of data semantics, etc. Following is the general style used by the data centers to perform data integration:

- **Data Integration Access**

In this style, the system processes the independent data source, which including semantic analysis; removes the duplication, data purification, transfer, and integration. Then the system saves the data in a database, to provide a unified data set to the users [5]. The integrated data center independent with operational data source.

- **Distributed Real-time Union View**

This model allows the distributed data to be associated into a greater and applicable meaningful reasonable view. The data were located in a variety of data sources and data sheets, which were distributed according to their location. The access of data’s users comes from a remote data source directly. So when the user accesses the data, the data center integrates the data source and then replies to the users with the appropriate result [9]. That means the model can provide users with real-time integrated and consistent data.

- **Combining Data Integration with Data Union (Disposition Plan)**

This model fixed the problem and conducts the integration process by unified the tow former schemes in special situations where the application systems providing a data source in different environments and at the same time these systems are online. So it provides dynamical data integration mode which combines data integration with data union. The data integration mode used when the system was first built toward online and off line. Then and after integration the data should be written into the center database [10]. The advantages and disadvantages of each model are shown in table 1.

TABLE 1: Advantages and disadvantages of the three studied models.

The model	Advantages	Disadvantages
• Data integration access model.	<ol style="list-style-type: none"> 1. The possibility of performing an integration process at once. 2. Fast and easy to emerge and manage the user’s calling process. 	<ol style="list-style-type: none"> 1. It is unsuitable to provide the data source for on-line system. 2. The modification of data such as insertion and deletion cannot be reflected in the integrated data source in real time.
• Distributed real time union view.	<ol style="list-style-type: none"> 1. The ability to provide a user with real-time data source. 	<ol style="list-style-type: none"> 1. Increasing of repose time based on the internet speed and the location of the calling data source.
• Combining Data Integration with Data Union (Disposition Plan)	<ol style="list-style-type: none"> 1. Tackle the system in special situation (online systems working in different environments). 	<ol style="list-style-type: none"> 1. The difficulties of applying the model.

B. The mechanisms of data integration in data centers

The aim of the data center is to integrate the isolated data source which is dispersed in a different site. During the integration, the data is attached to the database developed in distinctive platforms for different purposes; as a result the data structure is always unlike. The development environment, data structure, redundancy of data and the coupling of systems all together deciding the complexity of integration and where the promise of integration is to avoid affecting the normal system's operations. There are many mechanisms used to collect the data from the different data sources, the following section discusses the most common mechanisms.

- **The Data Centre's Active Pull**

In this model, based on the requirements and during the time where the system is the idle, center pulls the data from different data source. Then process, integrate, and write the data in the central database. The feature of this mode is that the Data Center has initiative which is good for balancing the burden of the data center. On the other hand, the data center has to understand the status of data source and the update of the data. This certainly enhances the coupling between the data sources and increases the processing burden of the data center [11].

- **Data Source's Active Push**

This model work in an opposite way of the previous one, here the initiative is from the data source it calls the service interface pushes the data to the data center in case of update. The role of the center is to process and integrate the data then write it to the data center database which released a huge process burden from the data center [12].

- **Data Source Active Push Combining with Data Centre Active Pull**

This model combines both former models (active pull, active push). The active pull mode is used at the beginning of building the data center in order to construct the basic center database, and later in the stage of system maintenance and running the data source active push model used to complete the update and enrich the center database to ensure provide the users real-time, integrated and consistent data [13].

4. DATA CENTER TECHNIQUE AND ARCHITECTURE

The rapid development of the computer and the communication technology, lead to fast enhancement in the information industry, which led to increase the information systems used by organizations. Although all these information systems have the same type of data but each has its own database which leads to redundancies and data inconsistency [14].

While the long-established technology such as DCOM +, CORBA, J2EE integrates the distributed and heterogeneous systems, but these technologies show their limitations from the point of maintaining a certain degree of automatic interaction. So a lot of solutions have been proposed to tackle these problems and overcome these limitations following is a discussion of three of most used architecture built using different technologies.

A. Multi-Agent's social

Multi-Agent's social (MAS) one of the proposed architecture that had a lot of enhancement such as initiative and autonomy, mobility, and other features that can help to facilitate the enterprise information integration. Using the MAS the enterprise distributed systems can be considered as agents, with the ability of interaction while these agents are moving [15]. In order to increase the flexibility the MAS technology has been built based on three layers User, Business and Model layer (UBM) as depicted in figure 2.

- **User layer**

This layer includes two types of users the first one is the end user whom interact directly with the system through the user interface, the second type of user is applications, which utilizing the application interface to access the datacenter [16].

- **Business layer**

Large enterprise applications are often structured around the concepts of business processes and business components. These concepts are addressed through a number of components, entities, agents and interfaces in the business layer. Business components are the software realization of business concepts. They are the primary units of design, implementation, deployment, maintenance and management of the life cycle of the business application [15].

- **Model Layer**

Most business applications must access data that is stored in corporate databases, which are most often relational databases. Model layer includes the different data source, existing systems, and their interfaces. Data access components in this model layer are responsible for exposing the data stored in these databases to the business layer [17].

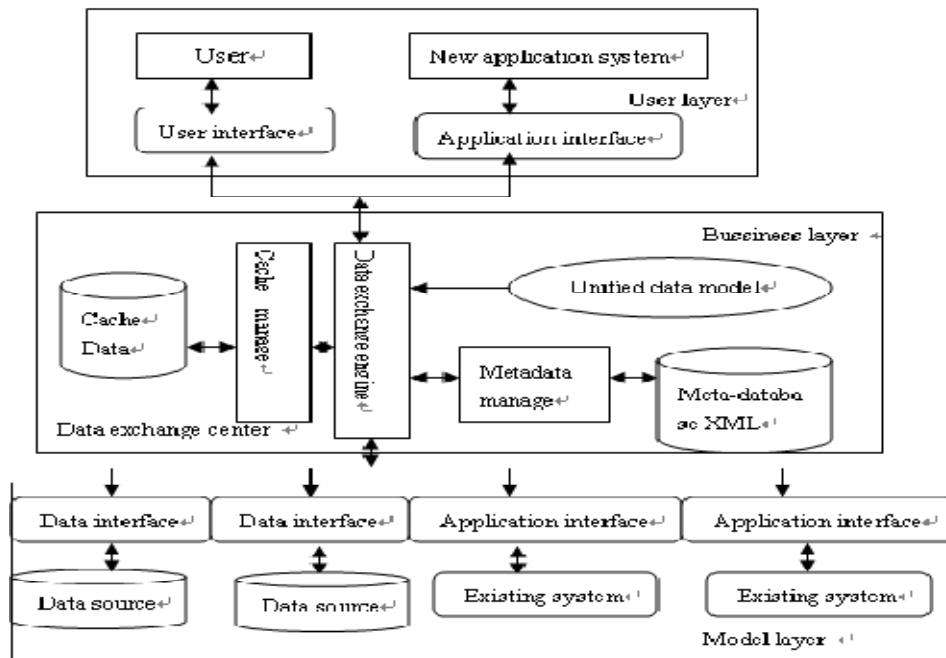


FIGURE 2: U-B-M the three-layer information integration platform architecture.

Although the MAS model helped to improve the data integration between the different and isolated systems, but it has some limitation such as the inability to provide full loose coupling, working on three layers only bounds the flexibility of the model [17] so it increases the required efforts to perform the integration.

B. Data Integration based on CORBA and XML

The main problem of CORBA data-integration blueprint is strictly coupled interface to the data source structure which means any change of data source structure will cause the interface rewriting. Using XML to realize the data source of outsourcing was a good solution, but the problem of XML data integration blueprint is that XML does not permit any way of using the distribution objects directly. It can't send a request to programmable language service but CORBA can resolve this problem and provide a public service such as event service, message service. So this technique tries to overcome the limitations CORBA and XML by combining them together. Besides, the CORBA technology is the basis of building distributed object system, fulfilling the development way based on component, promoting software reuse and the integration of legacy system. As shown in figure 3 the fundamental mind of this structure is: CORBA is the middleware of system integration and the component adhesives [18]. The CORBA can realize the intercommunication among components and shield distribution of various data sources and heterogeneous systems.

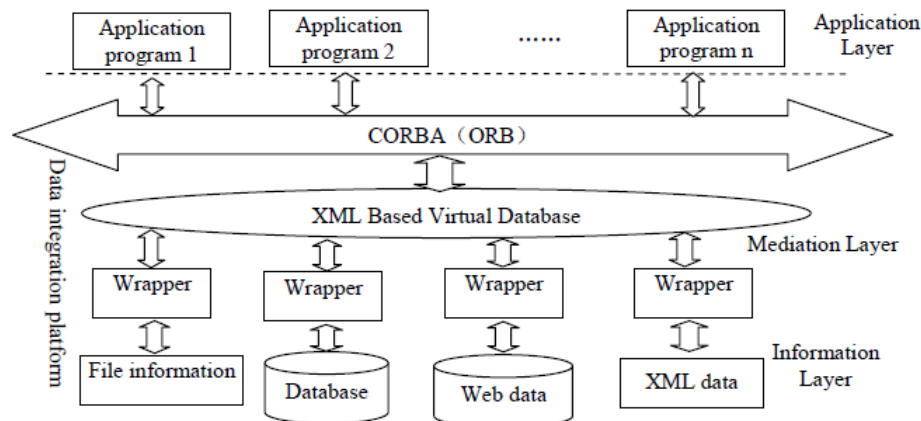


FIGURE 3: Data integration framework based on CORBA and XML [18].

The XML is used to describe the integrate data, the XSL is used to define the client view, the document XNIL and the format DTD are used to express the mapping between integrated pattern and data source. Various data source patterns integrate into global data patterns through corresponding wrapper. In this processing, the data in the data source is converted to an object DOM by the wrapper. Application layer: client interface layer, which is the same to the blueprint based on CORBA, develops the suitable application component. The development is based on the specific application and client request. To provide some flexibility this model contains four layers as follow:

- **Information layer**

The information layer being the bottom layer, which is the same to the above structures, is the data provider of the system. The layer contains all kinds of database, documents and multimedia and so on.

- **Mediation layer**

The layer is composed of a data integration platform and CORBA.

- **Application layer**

Client interface layer, which is the same to the blueprint based on CORBA, develops the suitable application component. The development is based on the specific application and client request.

C. *Service oriented architecture as techniques for data center*

Service oriented architecture is an architectural style whose goal is to achieve loose coupling among interacting software agents. While “service is a unit of work done by a service provider to achieve desired results for a service consumer”. Both provider and consumer are roles played by software agents on behalf of their owners [19].

SOA came to represent the vision of a composite computing model which is “An architecture that uses a distributed, discovery-based execution environment to expose and manage a collection of service-oriented software assets.” The software assets not more than a part of business logic; it can be a piece of code that performs a useful business function we want to provide to the outside world [20]. The composite computing model has been designed to fulfill some business requirements such as:

- To provide a high level of separation of concern between the capabilities of business logic’s and its implementation.
- Dynamic service systems, automatic discovery and composition of business logic’s capabilities.
- Quickly assemble unrehearsed computing communities with negligible coordinated planning efforts, installation procedures, or human involvement.

Figure 3 shows how a Java application can interact and invoke the function provided by ASP.NET server using the web service description language (WSDL) as unified interface.

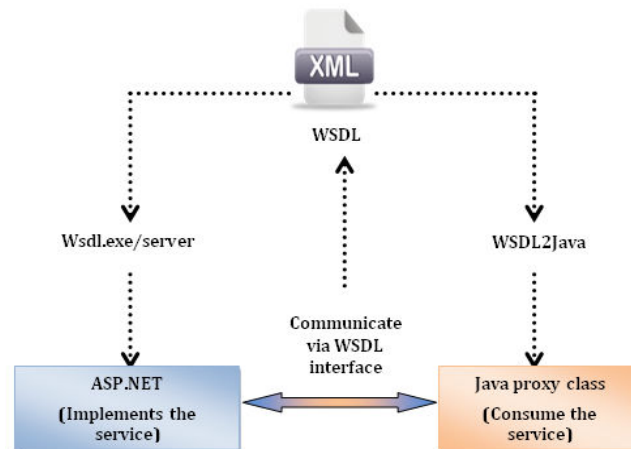


FIGURE 4: The role of WSDL in different agent interaction [19].

Service oriented architecture uses to integrate a various information systems in the organization and creating heterogeneous environment. As shown in figure 5 there are two views of the functional service requirement [20]. The provider view which is the business and technical capability that a service must deliver to satisfy its consumers, while the consumer view is the business and technical capability that the service is expected to deliver in the context of that consumer alone. In order to perform the data integration the same application may have the both views and can play the two different roles (data provide and data consumer) by consuming and providing a different service interfaces [4]. Figure 5 shows an SOA as a set of logical layers that are relatively independent, which lets an organization choose the degree of consumer-provider integration.

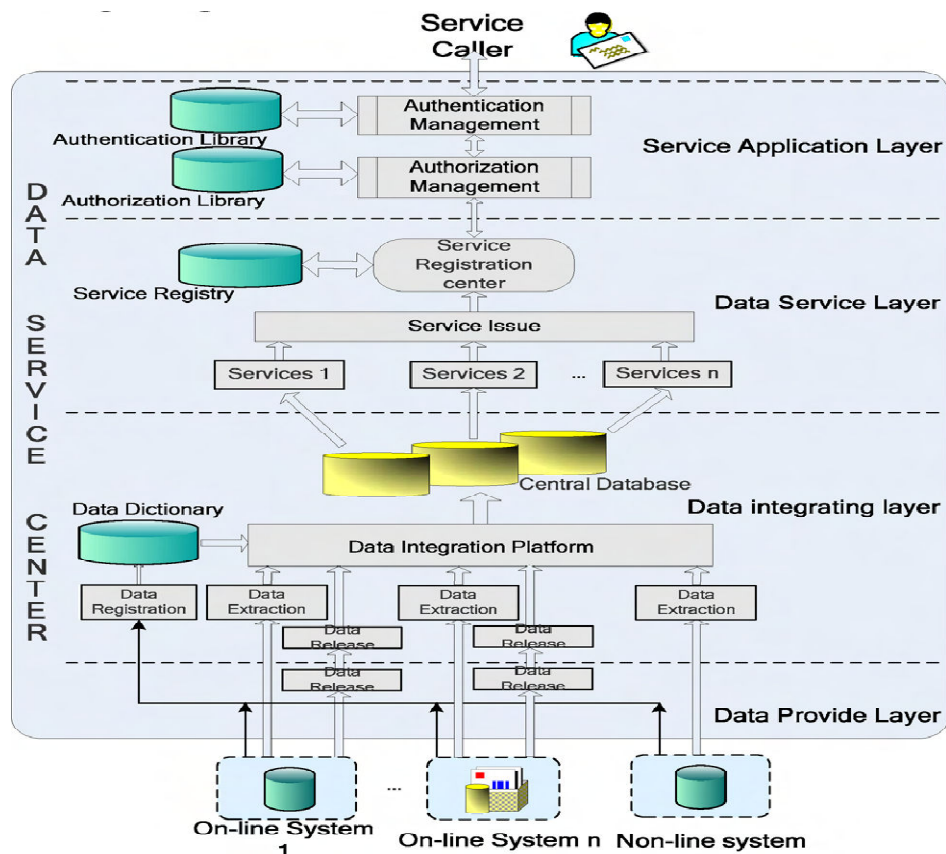


FIGURE 5: Data center architecture based on SOA [4]

- **Data Provider Layer**

This layer is the contact layer with the integrated systems that provide the data source. This layer including tow type of data, structured data such as (OB2, Access, Oracle , SQL Server, and so on) this data accessed by database management systems, unstructured data accessed as various OS file forms.

- **Data Integrating Layer**

In addition to management and maintaining the database, this is also layer is responsible about checking the syntax and the semantics of the data. And constructing the dictionary will be to describe the database constructor. Then the data will be integrated and fused to the center database to be ready to call.

- **Data Service Layer**

Provide a unified resource management and unified data view and unified service calling interface and so on; supply the users the service orientated duty and rules. Shield the differences of structure and usage of underlying resources. Encapsulate, register and publish all kinds of data and construct relatively independent service calling interface

- **Service Application Layer**

This layer exposes the data service and makes it available to the client systems. It is responsible for the management about user authentication and security call. All the calling of SOA must be authorized by safety certification center and was audited. Safety certification center configures Access source Matrix to control the behavior of different authorized users.

The principle of the central design (Object oriented systems) is to separate the interface from the implementation, in other words, the variation on independence between the object's interface and its implementation. This principle facilitates management of dependencies between objects by enforcing the predefined interface as the only way of interacting which facilitate the integration with other systems in the future [21].

- Since the main problem of software development and integration is the continuous change in the working environment, increasing user requirements, and managing the complexity of large software. The solution was not to develop solid software that anticipates the future requirement and doesn't need for any changes, while that is impossible [22]. So the best solution is to develop very flexible software can be easily updated to include the emergence of user requirement and the environment changes.

- Although there are many methods and technique to solve this problem on the level of programming such as structured programming, functional programming, stepwise tenement, logical programming, and object-oriented programming oriented programming and usage of classes. Furthermore, introduce more levels of abstraction was one of the attempts to master the complexity at the level of development. Usage module was the most effective and successful approach to handling the complexity to the design level [23] [24].

- Using different service for non functional requirement such as authorization and authentication provide a level of separation between the concern which facilitate the process of management and data exchange.

Despite of all of these advantages of SOA as a data integration technique but it still have some limitation such as:

- **Business Service Governance & Auditing**, As it mentions earlier business service are involving many agents such as service providers, service consumer, and the register (Broker), this made the process of defining the policies and managing this complexity and often dynamic relationships is very challenging [25].

- **Service Level Compliance**, Business services are comprised of different Web services and these often depend on other Web services. All of these may have inconsistent service level objectives. Service-level compliance of business services creates new requirements for performance and availability policy definition, reporting and troubleshooting [26].

- **Business Service Lifecycle Management**, business services will require a highly collaborative life cycle among all constituencies within a service delivery chain to optimize the flexibility and agility afforded by an SOA. Business people, developers and operators will need to have the same information available to them but within their own context [26].

From the literature we can conclude that there are many factors determining the barriers and challenges that can faced by the organization to perform the data integration. Furthermore, there was no single model, technique or mechanism can suit all the different situations. Before determining the integration architecture a deep analysis must be conducted regarding environment, data architecture, business process and the goal of integration. But as general we can notice breaking down the software in different layers has a lot of advantages to facilitate the integration "SOA". Table 2 summarizes the different challenges of data integration and the most suitable architecture can overcome these challenges.

TABLE 2: data integration challenges and the suite data integration architecture.

Data center architecture Data integration challenges	Multi-Agent's social	Data Integration based on CORBA and XML	Data Integration based SOA
Application rollout downtime		✓	✓
Running multiple versions	✓	✓	
Constrained interface development		✓	✓
Writing data access code		✓	✓
Data synchronization	✓	✓	✓
Aging, legacy integration engine technology	✓	✓	✓
Interface volumes are increasing significantly			✓
Growing demand and the complexity of heterogeneous IT environments			✓

6. CONCLUSION

The paper started by demonstrating the importance of data integration and the challenges facing by organizations in achieving this goal. It clearly demonstrates the importance of dealing with data integration for organizations and the benefits of data centers as the mechanism to facilitate data integration between organization. Furthermore, this paper reveals some areas where studies on data integration were conducted. In additional it discussed different kind of technologies and supported by analysis on the strength and weaknesses of the available software architectures for data integration. Finally the paper clarified the benefit and the general idea behind the SOA, which found to be the most suitable architecture for data integration in heterogeneous environment. Future work of this study would be, to conduct an empirical study to determine the strength of SOA and the suitability of its architecture as the technique for data integration.

REFERENCES

- [1] Shaw, M. a. D. G. (1996). "Software Architecture: Perspectives on an Emerging Discipline." Prentice-Hall.
- [2] Dave, h. "Common Models in SO A Tackling the Data Integration Problem" Progress Software Corporation 2008.
- [3] NASDAQ. "data integration challenges with SOA" (2011) available at www.progress.com.
- [4] Shuiping Zhang, J. L., Fengqin Zhang, Xiaobo Li (2011). "Design of Data Center and Implementation of Data Integration Based on SOA Framework." IEEE".
- [5] Chunhua Gu, X. Z. (2010). "An SOA Based Enterprise Application Integration Approach." 3rd International Symposium on Electronic Commerce and Security (ISECS 2010), pp. 324-327.
- [6] Min-Yuh Day, C.-S. O., Wen-Lian Hsu (Aug 2009). "An analysis of research on Information Reuse and Integration." IEEE International Conference on Information Reuse & Integration (IRI 2009): 5.
- [7] P. Ziegler. User-Specific Semantic Integration of Heterogeneous Data: What Remains to be Done? Technical Report ifi-2004.01, Department of Informatics,, University of Zurich. <http://www.ifi.unizh.ch/techreports>.
- [8] Lynne A. Dunbrack, Marc H. "The Critical Need for Connected Healthcare: Meeting Today's Integration Challenge" Health Industry Insights USA available at "www.healthindustry-insights.com" 2008.
- [9] Zhuofeng Zhao, J. F., Jing Cheng, (2008). "CAFISE-S: An Approach to Deploying SOA in Scientific Information Integration." 2008 IEEE International Conference on Web Services (ICWS 2008): pp. 425-432, S.

- [10] Wang Panqing, L. X., Liu Zengliang (2010). "The design and implementation of information integration framework on equipment domain based on ontology," 2010 May 2010, doi 10.1109/IICNDS.2010.5479191." 2nd International Conference on Networking and Digital Society (ICNDS 2010): pp. 310-313.
- [11] Peng Gong, C. Y., Dongping Shi (April 2007). "Heterogeneous information processing and integration method research in Enterprise Management Information Systems." Proceedings of the 2007 11 th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2007): pp. 951-955, 926-928.
- [12] Xiangyu Li (2010). "A Multi-layer Information Integration Framework for Process Operation System." 2010 Third International Conference on Information and Computing (ICIC 2010), : pp. 3-6.
- [13] Patiwat Wisedsukol, P. C., Kittivong Subphachaisirikul, (July 2010). "Heterogeneous Information Integration and Management Models in Thai Justice Information Systems." 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010), volA: pp. 269-273.
- [14] Harrison Colin G, C. D. M., Kershenbaum Aaron (2003). "Mobile Agents : are they a good idea ? Research Report " IBM Research Division.[EB/OL].www.research.ibm.com/massive/mobag.ps.19952052.
- [15] Li Yang, G. Z., Xinyu Geng, Haiyan Yuan (2010). "Research on Virtual Data Center Architecture Based on Intelligent Multi-agent." IEEE 2010 International Conference on Computational and Information Sciences.
- [16] Blazona, B., Koncar, M.: HL7 and DICOM based integration of radiology departments with healthcare enterprise information systems. International Journal of Medical Informatics 76, 425-432 (2007).
- [17] "Microsoft Corporation. Application Architecture for .NET: Designing Applications and Services." (2003.) Available at:<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/distapp.asp>.
- [18] Yang M., Zhang Y. "Quality information system data integration based on CORBA and XML" (2009) IEEE
- [19] He, H. (2003). "What Is Service-Oriented Architecture." <http://www.xml.com/pub/a/ws/2003/09/30/soa.html>.
- [20] Arsanjani A., Ellis M., Abdul Allam, Kishore C. (2007). "S3: Service-Oriented Reference Architecture." IEEE Computer Society.
- [21] Gartner. Newman D. "Data Integration: At the Heart of the Data Services Evolution", June 11–13, 2007, Gaylord Opryland Resort & Convention Center, Nashville, TN.
- [22] Imran Sarwar Bajwa, R. K., Shahzad Mumtaz, M. Abbas Choudhary, and M. Shahid Naweed (2008). "SOA and BPM Partnership: A paradigm for Dynamic and Flexible Process and I.T. Management." World Academy of Science, Engineering and Technology.
- [23] Elgedawy, I. (2009). "Reusable SOA Assets Identification Using E-Business Patterns." 2009 World Conference on Services - II, IEEE.
- [24] Eun-Ju Park, H.-K. K., Roger Y. Lee. (2007). "Frameworks of Integration Repository for Software Process Improvement using SOA." 6th IEEE/ACIS International Conference on Computer and Information Science.
- [25] Amelia Maurizio, J. S., Peter Jones, Gail Corbitt and Lou Girolami (2008). "Service Oriented Architecture: Challenges for Business and Academia." Proceedings of the 41st Hawaii International Conference on System Sciences - 2008.
- [26] Chappell, D. (2002). "Java Web Services." O'Reilly.